Fiscal Year Report

Principal Investigators: Drew McDermott and Gregory Hager
Institution: Yale University Department of Computer Science
Phone: (203) 432-1997
E-mail: mcdermott@cs.yale.edu
Title: Knowledge-Based Planning
Grant Number: N00014-91-J-1577
Reporting Period: 1 Oct 1991 - 30 Sep 1992

1 List of Numerical Productivity Measures

- Refereed papers submitted but not yet published: 2
- Refereed papers published: 7
- Unrefereed reports and articles: 2
- Books or parts thereof submitted but not yet published: 0
- Books or parts thereof published: 0
- Patents filed but not yet granted: 0
- Patents granted (include software copyrights): 0
- Invited presentations: 12
- Contributed presentations: 0
- Honors received : 1 (Drew McDermott was Conference Chair of the First International Conference on AI Planning Systems)
- Prizes or awards received (Nobel, Japan, Turing, etc.): also include descriptions of the specific prizes. 0
- Promotions obtained: 0
- Graduate students supported $\geq$ 25% of full time: 2
- Post-docs supported $\geq$ 25% of full time: 0
- Minorities supported (include Blacks, Hispanics, American Indians and other native Americans such as Aleuts, Pacific Islanders, etc.; do not include Asians or Asian-Americans): 0

Fiscal Year Report

Principal Investigators: Drew McDermott and Gregory Hager
Institution: Yale University Department of Computer Science
Phone: (203) 432-1997
E-mail: mcdermott@cs.yale.edu
Title: Knowledge-Based Planning
Grant Number: N00014-91-J-1577
Reporting Period: 1 Oct 1991 – 30 Sep 1992

2 Summary of Technical Results

The goal of our project is to study planning for autonomous agents with imperfect sensors in a dynamic world. Such agents must confront several problems:

- how to synchronize plan execution with plan refinement
- how to generate reasonable plans quickly for complex goals, and improve them later
- how to trade off sensor-processing time against the quality of information
- and how to learn the structure of the environment as plan execution proceeds.

We have results in all these areas.

2.1 Transformational Planning of Reactive Behavior

In complex, dynamic worlds, plans must include sensing operations, and an agent does not have time to make up such complex plans from scratch when a goal pops up. The alternative is to provide a set of modular plans that can cope with most eventualities, and then paste these modules together to handle whatever constellation of goals arises. This planning algorithm, if you want to call it that, is fast, but prone to producing inefficient plans. Hence, simultaneously with the execution of the system's default plan, an off-line planner attempts to find an improved version. It *projects* the plan to generate sample execution traces, runs *critics* that look for standard "bugs" for the domain, and tries *plan transformations* to attempt to ameliorate those bugs.

In the past year, we have developed several technical results within this framework. For example, when an improved plan is found, the current plan must be abandoned. Our model of plan swapping is that the agent simply discards the old plan and begins the new one. The new plan typically runs some sensing operations to orient itself, and avoids repeating work done by the old version. However, this model is prone to a particular type of bug, where changes in the agent's belief state as a result of actions may not be incorporated because the plan is discarded between the beginning of the action and the recording of the belief changes. We have eliminated this kind of bug by addition of an "evaporation protection" primitive into our plan language. This primitive forces some plan steps to be carried out even if the plan has been discarded.

It is important when a critic processes a plan projection that it be able to recover the state of all data structures at any point in a plan. We have developed an efficient algorithm for recording and time-stamping all side effects during a projection. During the criticism phase, the exact state of the agent can be reconstructed at any point. The time and space required are both linear in the number of side effects to the data structure.

Reactive plans involve a high degree of concurrency. Plans often request resources, such as effectors, and block while they wait for them to become available. Because the planner puts disparate modules together, and then transforms the resulting code, it must cope with the likelihood that the resulting plans will suffer from deadlocks, in which a cycle of tasks are waiting for each other to release resources. We have developed a clean combination of the *task* idea from AI planning, and the *process* idea from operating systems. A task is a commitment to carry something out, and often gets reduced to subtasks that jointly do the work. Some tasks correspond to processes. A subtask's process ought to inherit resources from its supertasks' processes, because it corresponds to a part of its supertasks, and will presumably advance them. Our process mechanism has this feature, and supports an efficient mechanism for pre-empting resources and breaking deadlocks.

One of our most important results is our model of declarative goals in reactive plans. A declarative goal is an explicit statement of a condition to be made true. Our model assumes that tasks are procedural, so we must provide a fast and faithful transition from the declarative language to the procedural one. Our method is to provide for each declarative goal a robust *default method* that will under normal circumstances achieve that goal, possibly with less than optimal efficiency. For example, the default method for the goal $\forall x(P(x) \supset Q(x))$ is to find an object b that has property P but not property Q, and achieve $Q(b)$. Retrieving and instantiating default methods is quite fast, and gives the agent something useful to do while it attempts to find a better plan offline. Its search for a better plan is mediated by a set of transformations that refine particular patterns of goals into specialized actions. For example, if P mentions the location of an object, then there may be specific methods for going to a location and scanning for objects of particular sorts. These transformation rules are associated with decision-theoretic heuristics for judging whether they are likely to be effective given the behavior of the objects being manipulated.

2.2 Interval-Based Inference on Sensor Data

Many sensor-data-processing tasks can be be phrased in terms of finding values of parameters that satisfy given constraints. For example, determining whether a certain set of edges in an image could be an instance of an object model can be thought of as finding values for the object's parameters that account for the given edges. Our approach to finding such values is to start with intervals containing the correct values, and gradually prune away subintervals that are inconsistent, until we are left with subintervals that are guaranteed to contain at least one point satisfying the constraints.

In the last year, we have made the following advances:

- Implemented a distributed version of the algorithms. The resulting implementation is significantly faster (i.e. parallel) and more fault-tolerant.
- Implemented a data selection technique that reduced the time needed to solve some benchmark problems to less than 1 second (essentially real-time operation).
- Implemented and tested the algorithms on data requiring descrimination and comparisons among multiple objects or targets.
- Implemented a version of the algorithms for use in unstructured domains. This significantly increases the domain of applicability of the algorithms, making it possible, for example, to apply them outdoors.

We have also been using interval-based techniques for representing and constructing maps of the environment for mobile robots. Maps are represented as graphs of places, where each place is related to its neighbors by its relative location (restricted to a set of intervals) and the actions that will get the agent from one neighbor to another. In addition, stored with each place is a collection of *views,* which are visual signatures obtained at that place. Two places look alike if their signatures are sufficiently near. The key problem in map building is to decide when the agent is now at a place it has visited before. Our algorithm makes that decision based on overlaps of the current measure location and the stored location of the candidate matching location. An important innovation is that the algorithm continues to gather statistics on its identifications, and changes its mind when it thinks it must split previously merged places, merge previously distinctly places, or delete places that were erroneously observed in the first place.

Over the past year, we have developed a class of plans, called *exploration scripts,* for speeding up the pace of map building; the basic mapping algorithm does not require getting control of the robot, but can profit from an opportunity to do so. More recently, we have gathered data on what makes an image signature distinctive (less likely to match lots of places). The idea is to rotate the robot's camera until a distinctiveness peak is attained. The hope is that any given location will have few distinctiveness peaks, allowing us to store only a view fews. Preliminary experiments seem to show that this hope is reasonable.

Fiscal Year Report

Principal Investigators: Drew McDermott and Gregory Hager
Institution: Yale University Department of Computer Science
Phone: (203) 432-1997
E-mail: mcdermott@cs.yale.edu
Title: Knowledge-Based Planning
Grant Number: N00014-91-J-1577
Reporting Period: 1 Oct 1991 – 30 Sep 1992

3 Publications, Presentations and Reports

- Michael Beetz, July 1992, Panelist at the Panel "Planning and Scheduling" (Workshop "Implementing Temporal Reasoning," AAAI-92

- Michael Beetz, August 1992, Talk on "Improving and Debugging Reactive Plans that Contain Declarative Goals." German Research Center for Artificial Intelligence, Inc. (DFKI).

- Michael Beetz, August 1992, Talk on "Improving and Debugging Reactive Plans that Contain Declarative Goals." Bavarian Research Center for Knowledge-based Systems (FORWISS), Germany.

- Michael Beetz, August 1992, Talk on "Improving and Debugging Reactive Plans that Contain Declarative Goals." Technical University of Darmstadt, Germany.

- M. Beetz, M. Lindner, and J. Schneeberger 1992 Temporal Projection for Hierarchical, Partial-order Planning. *Proceedings of ISAI-92,* AAAI Press.

- Michael Beetz and Drew McDermott 1992 Declarative goals in reactive plans. In James Hendler (ed.) , *Proc. First Int. Conf. on AI Planning Systems,* San Mateo: Morgan Kaufmann, pp. 3–12

- Sean Engelson and Drew McDermott 1991 Image signatures for place recognition and map construction. SPIE Technical Symposium on Advances in Intelligent Robotic Systems.

- Sean Engelson and Drew McDermott 1992 Error correction in mobile robot map learning. *Proc. IEEE Conf. on Robotics and Automation*, pp. 2555–2560

- Sean P. Engelson and Drew McDermott 1992 "Active Place Recognition Using Image Signatures", to appear in Proceedings of SPIE Sensor Fusion V, November 1992

- Gregory D. Hager 1991 Towards geometric decision making in unstructured environments. In *Proc. 1991 International Workshop on Intelligent Robots and Systems*, Bellingham, WA, pp. 1412–1417.
- Gregory D. Hager 1992 "A Constraint-Based View of Selective Perception", Proceedings of the AAAI Spring Symposium on Selective Perception, Stanford, CA, March 1992.
- Gregory D. Hager 1992 "Constraint Solving Methods and Sensor-Based Decision Making" *Proc. IEEE Conf. on Robotics and Automation,*
- Gregory D. Hager, April 1992, "Sensor Data Fusion," a lecture delivered at Red Stone Arsenal, Huntsville, Alabama.
- Gregory D. Hager, June 1992, "Sensor-Based Decision Making" presented at the DLR (German Space Organization), Oberpfaffenhofen, Germany.
- Gregory D. Hager 1992 Task-directed computation of qualitative decisions from sensor data. Submitted for review to the *IEEE Transactions on Robotics and Automation.*
- Drew McDermott, Nov. 1991, Invited presentation on "Transformational Planning of Reactive Behavior" at Ohio State University.
- Drew McDermott, William Cheetham, and Bruce Pomeroy 1991 Cockpit emergency response: the problem of plan projection. *Proc. IEEE Conf. on Systems, Man, and Cybernetics*, Charlottesville, Virginia
- Drew McDermott 1992 "Perceptual Confusion in Reactive Plans," *Proc. of the AAAI Spring Symposium.*
- Drew McDermott, March 1992, Talk at Carnegie-Mellon on "Transformational Planning of Reactive Behavior."
- Drew McDermott, April 1992, Talk at University of Chicago on "Building and Fixing Diktiometric Maps for Robot Navigation."
- Drew McDermott, April 1992, Talk at Northwestern University, "Transformational Planning of Reactive Behavior."
- Drew McDermott, June 1992, Chaired panel on "Unified Theories of Planning," at the First International Conference on AI Planning Systems, U. of Maryland.
- Drew McDermott, July 1992, Seminar on "Classical and Reactive Planning," Bolzano Summer School, Italy.

Fiscal Year Report

Principal Investigators: Drew McDermott and Gregory Hager
Institution: Yale University Department of Computer Science
Phone: (203) 432-1997
E-mail: mcdermott@cs.yale.edu
Title: Knowledge-Based Planning
Grant Number: N00014-91-J-1577
Reporting Period: 1 Oct 1991 - 30 Sep 1992

4 Research Transitions And DoD Interactions

Prof. McDermott has been serving on the Technical Review Board for the DARPA Transportation and Scheduling Initiative. The purpose of the board is to provide high-level feedback to researchers in this area, using insights gained from past research on planning and scheduling.

Fiscal Year Report

Principal Investigators: Drew McDermott and Gregory Hager
Institution: Yale University Department of Computer Science
Phone: (203) 432-1997
E-mail: mcdermott@cs.yale.edu
Title: Knowledge-Based Planning
Grant Number: N00014-91-J-1577
Reporting Period: 1 Oct 1991 - 30 Sep 1992

5 Software And Hardware Prototypes

1. The Reactive Plan Language interpreter is available, and runs in Lucid Common Lisp. We are attempting to port it to the University of Washington, and other institutions.
2. We have recently finished a version of the constraint solving algorithms that we are giving to several interested institutions.